



(19) Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) DE 103 03 490 A1 2004.08.12

(12)

Offenlegungsschrift

(21) Aktenzeichen: 103 03 490.0

(22) Anmelddato: 30.01.2003

(43) Offenlegungsdatum: 12.08.2004

(51) Int Cl.?: **B60R 16/02**

G05B 19/042

(71) Anmelder:

Robert Bosch GmbH, 70469 Stuttgart, DE

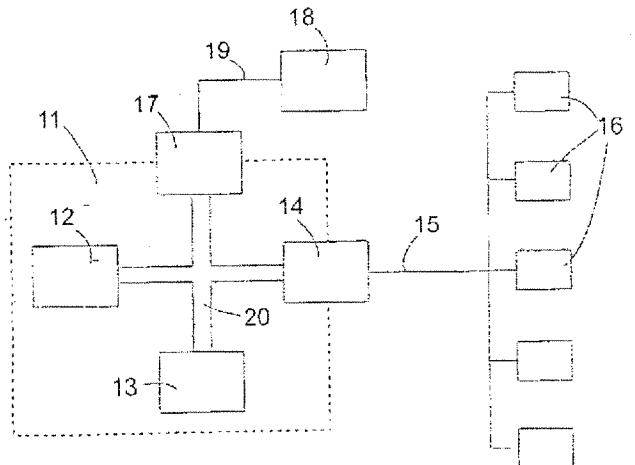
(72) Erfinder:

Laichinger, Martin, 73061 Ebersbach, DE; Aue, Axel, 70825 Korntal-Münchingen, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Steuergerät für ein Kraftfahrzeug und Kommunikationsverfahren dafür**

(57) Zusammenfassung: Ein Kraftfahrzeugsteuergerät, insbesondere ein Motorsteuergerät (11), umfasst einen Prozessor (12), eine erste Schnittstelle (14) für die Kommunikation mit Funktionseinheiten (16) des Kraftfahrzeugs und wenigstens eine zweite Schnittstelle (17), die mit dem Prozessor (12) in einer Baueinheit vereinigt ist.



Beschreibung

[0001] Die vorliegende Erfindung betrifft ein Steuergerät zum Steuern von Funktionen eines Kraftfahrzeugs, insbesondere des Motors eines solchen Kraftfahrzeugs wie etwa Kraftstoffeinspritzung, Zündzeitpunkt, etc. sowie ein Verfahren für die Kommunikation mit einem solchen Gerät. Ein solches Steuergerät verfügt im Allgemeinen über einen Prozessor, der, gesteuert durch ein Betriebsprogramm, Messwerte von Betriebsparametern des Kraftfahrzeugs von externen Funktionseinheiten empfängt und von diesen Messwerten Befehle zum Steuern der gleichen oder anderer Funktionseinheiten ableitet, die an die Funktionseinheiten übertragen werden.

Stand der Technik

[0002] Es ist bekannt, für die Kommunikation zwischen dem Steuergerät und den diversen Funktionseinheiten einen Bus, z.B. nach dem CAN-Standard, einzusetzen, an den das Steuergerät und die diversen Funktionseinheiten angeschlossen sind. Ein solcher Bus muss zum einen echtzeitfähig sein, d.h. er muss gewährleisten können, dass zwischen dem Aufnehmen eines Messwerts durch eine Funktionseinheit und dem Eintreffen eines in Reaktion auf den Messwert von dem Steuergerät erzeugten Steuerbefehl an der gleichen oder einer anderen Funktionseinheit eine vorgegebene Verzögerungszeit nicht überschritten wird. Außerdem muss die Übertragung auf dem Bus wirksam gegen Übertragungsfehler abgesichert sein, um Fehlsteuerungen zu vermeiden. Diese Anforderungen beschränken die auf dem Bus erreichbaren Übertragungsraten.

[0003] Bei der Neuentwicklung von Steuergeräten stellt sich das Problem, dass wesentlich größere Datenmengen mit dem Steuergerät ausgetauscht werden müssen, als dies beim späteren Einsatz des fertigen Steuergeräts in einem Kraftfahrzeug der Fall ist. Diese großen Datenmengen kommen dadurch zu stande, dass das Steuergerät nicht nur mit den Funktionseinheiten des Motors (oder mit Emulatoren, die sich gegenüber dem Steuergerät als diese Funktionseinheiten ausgeben) kommunizieren muss, sondern zusätzlich noch mit einem externen Entwicklungstool, das es einem Entwickler ermöglicht, die Abarbeitung des Steuerprogramms durch den Prozessor detailliert zu verfolgen und zu beeinflussen. Wenn die Kommunikation mit dem Entwicklungstool auf dem gleichen Bus stattfindet, über den das Steuergerät mit den Funktionseinheiten kommuniziert, ergibt sich das Problem, dass dessen Übertragungskapazität durch die Kommunikation mit den Funktionseinheiten bereits weitgehend ausgeschöpft ist und für die Kommunikation mit dem Entwicklungstool nicht mehr ausreichend Kapazität zur Verfügung steht. Eine daraus resultierende Verzögerung der Datenübertragung zum Entwicklungstool kann fälschlicherweise den Eindruck einer Fehlfunktion des Steuerge-

räts hervorrufen. Zwar ließe sich die Übertragung an das Entwicklungstool beschleunigen, in dem diesem auf dem Bus eine ausreichend hohe Priorität zugeschenkt wird, doch führt dies wiederum dazu, dass Anforderungen an das Echtzeitverhalten der Kommunikation zwischen Steuergerät und Funktionseinheiten nicht zuverlässig erfüllt werden können.

Aufgabenstellung

[0004] Ein Ansatz zur Lösung dieses Problems ist die so genannte ETK-(Emulator-Tast-Knopf-)Technologie. Bei dieser wird für die Entwicklung und Optimierung eines neuen Steuergeräts ein spezieller Prototyp eingesetzt, der sich von dem später in Serie hergestellten Gerät dadurch unterscheidet, dass eine Vielzahl von Anschlüssen herausgeführt ist, die es einem aufgesetzten Tastknopf erlauben, auf den internen Datenbus des Steuergeräts zuzugreifen, und so einen Speicher des Steuergeräts zu lesen und zu beschreiben, etc.

[0005] Diese speziellen Prototypen sind kostspielig, und die herausgeführten Anschlüsse belasten elektrisch die Steuergeräteschaltung, so dass Unterschiede im zeitlichen Verhalten und in den elektrischen Eigenschaften zwischen Prototyp und Serienmodell nicht völlig auszuschließen sind.

Vorteile der Erfindung

[0006] Die vorliegende Erfindung löst das Problem der beschränkten Übertragungskapazität, in dem sie dem Prozessor des Steuergeräts eine zweite Schnittstelle zur Verfügung stellt, die ausschließlich für die Kommunikation mit einem Debugger oder anderen externen Entwicklungstools nutzbar ist. Über diese Schnittstelle können beliebige Informationen zwischen dem Steuergerät und dem Entwicklungstool ausgetauscht werden, ohne dass dadurch das Zeitverhalten bei der Kommunikation über die erste Schnittstelle beeinträchtigt wird. Da die Schnittstelle mit dem Prozessor des Steuergeräts in einer Baueinheit integriert ist, können die Leitungen zwischen dem Prozessor und der Schnittstelle kurz gehalten werden, so dass die kapazitive Belastung der Schaltkreise des Steuergeräts durch diese Schnittstelle gering ist.

[0007] Um große Datenmengen in kurzer Zeit über die zweite Schnittstelle übertragen zu können, ist diese vorzugsweise für den Blocktransfer von Daten ausgelegt.

[0008] Bei der Baueinheit, in der der Prozessor und die zweite Schnittstelle vereinigt sind, kann es sich um eine Leiterplatte handeln, auf der der Prozessor und die zweite Schnittstelle jeweils einen Chip oder Chipsätze darstellen, vorzugsweise handelt es sich jedoch um einen einzigen Chip, in welchem der Prozessor und die zweite Schnittstelle integriert sind.

[0009] Um die Zahl der aus der Baueinheit herausgeführten Anschlüsse einer solchen Baueinheit ge-

ring zu halten, ist die zweite Schnittstelle vorzugsweise eine serielle Schnittstelle. Es kann sich z.B. um eine Internet-, Fire-Wire- oder, besonders bevorzugt, um eine USB-Schnittstelle handeln.

[0010] Die erste Schnittstelle ist vorzugsweise ebenfalls mit der zweiten Schnittstelle und dem Prozessor in der gleichen Baueinheit vereinigt. So kann darauf verzichtet werden, einen lokalen Bus für die Kommunikation des Prozessors mit der ersten Schnittstelle aus der Baueinheit herauszuführen.

[0011] Ein wichtiger Vorteil der USB-Schnittstelle ist ihre Flexibilität bei gleichzeitig einfacherem, preiswertem Aufbau. Aufwendige Verwaltungsfunktionen sind am Host des USB-Busses angesiedelt, d.h. bei der hier betrachteten Anwendung im Entwicklungstool. Am Steuergerät selbst ist nur einfache Hardware erforderlich. Da aber der USB kein Multi-Master-Bus ist, sondern sämtliche Übertragungsvorgänge vom Host ausgelöst werden müssen, bedarf es beim Einsatz einer USB-Schnittstelle als zweite Schnittstelle für die Kommunikation mit dem Entwicklungstool der Aushandlung einiger Besonderheiten des USB.

[0012] Eine wichtige Eigenschaft der USB-Schnittstelle ist ihre Fähigkeit, unterschiedliche Übertragungsmodi für die Übertragung zu unterschiedlichen Endpoints zu unterstützen. Insbesondere stehen Blockmodi zur Verfügung, um größere Datenmengen ohne Busoverhead zu übertragen. So können unterschiedlichen Übertragungsaufgaben zwischen Entwicklungstool und Steuergerät wie etwa Messen von Werten, Übertragung von Bypassbotschaften vom oder zum Steuergerät, Ändern von Datenwerten im Speicher des Steuergeräts oder Debugging des Codes des Steuergeräts jeweils eigene Endpoints und über diese jeweils spezifische Prioritäten und an die Anforderungen der Übertragungsaufgabe hinsichtlich Zeitverhalten und Kapazität angepasste Modi zugeordnet werden. So ist z.B. von Vorteil, wenn Bypassbotschaften, die eine hohe Priorität haben müssen, über einen für sie spezifischen Endpoint im Bulk- oder Isochrone-Transfermodus übertragen werden. Um diese Botschaften vorrangig zu verarbeiten, kann das Entwicklungstool so lange ausschließlich deren Endpoints abfragen, bis diese Botschaften abgearbeitet sind. Ein weiterer Vorteil ist die Möglichkeit der Übertragung von hohen Datenraten in eine Übertragungsrichtung. So können z.B. die Messwerte einer konfigurierten Messung mit garantierter Übertragungsbandbreite im Isochrone-Modus mit einer vom Entwicklungstool spezifizierten Paketgröße von bis zu 1023 Bytes übertragen werden. Das Entwicklungstool kann so bereits bei der Konfiguration der Messung die benötigte Busbandbreite reservieren und dadurch sicherstellen, dass durch die Anwesenheit anderer Busteilnehmer (d.h. durch Übertragung von und zu anderen Endpoints) keine Daten verloren gehen.

[0013] Das erfindungsgemäße Kraftfahrzeugsteuergerät kann ausschließlich als Prototyp für die Entwicklung eines neuen Steuergerätes eingesetzt wer-

den, das in Serie ohne die zweite Schnittstelle zum Einsatz kommt. Da aber insbesondere bei Realisierung auf dem gleichen Halbleiterchip die mit der Hinzufügung der zweiten Schnittstelle verbundenen Kosten sehr gering sind, kann ein solches Steuergerät auch serienmäßig in ein Kraftfahrzeug eingebaut werden.

[0014] Dies hat den weiteren Vorteil, dass mit Hilfe der zweiten Schnittstelle auch am fertigen Fahrzeug im Steuergerät gespeicherte Betriebsparameter auf einfache Weise ausgelesen werden können. Dadurch können Funktionen, die herkömmlicherweise z.B. von kostspieligen Fahrtenschreibern ausgeführt werden, einfach und preiswert in ein erfindungsgemäßes Steuergerät integriert werden.

[0015] Zudem ist eine einfache Möglichkeit geschaffen, Fehlerzustände zu messen bzw. zu diagnostizieren. Insbesondere könnte ein handelsüblicher PC zu Diagnosezwecken direkt mit dem Steuergerät kommunizieren.

[0016] Weitere Merkmale und Vorteile der Erfindung ergeben sich aus der nachfolgenden Beschreibung eines Ausführungsbeispiels mit Bezug auf die beigefügten Figuren.

Figuren

[0017] Es zeigen:

[0018] **Fig. 1** ein Blockdiagramm eines herkömmlichen Kraftfahrzeug-Steuergeräts; und

[0019] **Fig. 2** ein Blockdiagramm eines erfindungsgemäßen Steuergeräts.

Beschreibung der Ausführungsbeispiele

[0020] Um die Besonderheiten des erfindungsgemäßen Steuergeräts deutlich zu machen, wird eine kurze Beschreibung eines herkömmlichen Steuergeräts anhand der **Fig. 1** vorausgeschickt. Dieses Steuergerät 1 umfasst einen Mikroprozessor 2, einen Speicherbaustein 3 und eine Schnittstelle 4 zu einem Bus, insbesondere einem CAN-Bus 5, an den eine Mehrzahl von Funktionseinheiten 6 des Kraftfahrzeugs angeschlossen sind, welche Messwerte von Betriebsparametern des Fahrzeugs über den Bus 5 an das Steuergerät 1 liefern und/oder Befehle vom Steuergerät 1 empfangen und ausführen. Ein Beispiel für die Funktionseinheiten 6 sind Drehwinkelauflnehmer an einer Kurbelwelle des Fahrzeugmotors, die einen Messwert für den Drehwinkel der Kurbelwelle an das Steuergerät 1 liefern, und Zündkerzen des Motors, die einen vom Steuergerät 1 aus den Messwerten der Drehwinkelauflnehmer abgeleiteten Zündbefehl empfangen. Eine Funktionseinheit 6 kann auch ein Dosiersystem für die Kraftstoffeinspritzung in den Motor sein, das vom Steuergerät angesteuert wird, um während eines Getriebe-Schaltvorgangs die Einspritzmenge kurzzeitig zu reduzieren.

[0021] Die Funktionseinheiten können noch diverse andere Mess- und Regelungsaufgaben erfüllen, die

hier nicht im Detail beschrieben werden.

[0022] Während sich das Steuergerät 1 noch in der Phase der Erprobung und Entwicklung befindet, ist an den CAN-Bus 5 eine Schnittstelle 7 angeschlossen, über die ein auf einem externen Computer implementiertes Entwicklungstool 18 mit dem Steuergerät 1 kommunizieren kann.

[0023] Bei dem in **Fig. 2** gezeigten erfindungsgemäßen Steuergerät 11 ist ein Mikroprozessor 12 mit einem Speicherbaustein 13 und zwei Schnittstellen 14, 17 über einen internen Adressdaten- und Steuerbus 20 verbunden. Da die Leitungen des internen Busses 20 kurz sind – wenn das Steuergerät aus einer Mehrzahl von auf einer Leiterplatte angeordneten integrierten Schaltungsbausteinen aufgebaut ist, wird die Leitungslänge des internen Busses ca. 10 cm im Allgemeinen nicht überschreiten; wenn die Komponenten 12, 13, 14, 17 auf einem gemeinsamen Halbleitersubstrat integriert sind, liegt die Länge der Busleitungen höchstens im Bereich von wenigen Millimetern – sind auf dem internen Bus 20 hohe Taktraten erreichbar, die es ermöglichen, beide Schnittstellen 14, 17 bis an die Grenze ihrer Übertragungskapazität auszulasten. Das Zeitverhalten der Kommunikation des Mikroprozessors 12 über den seriellen Bus 15 mit den externen Funktionseinheiten 16 ist daher unabhängig vom Umfang der Kommunikation zwischen dem Mikroprozessor 12 und dem an die Schnittstelle 17 angeschlossenen Entwicklungstool 18. Eine Verfälschung des Echtzeitverhaltens des Steuergeräts 11 durch das angeschlossenen Entwicklungstools 18 wird so vermieden.

[0024] Der externe Bus 19, der die Schnittstelle 17 mit dem Entwicklungstool 18 verbindet, ist ein serieller Bus. Der Wunsch, eine hohe Übertragungskapazität zwischen Steuergerät 11 und Entwicklungstool 18 zu erreichen, könnte zwar dazu veranlassen, einen parallelen Bus einzusetzen, die Verwendung eines seriellen Busses ist jedoch bevorzugt, weil sie es ermöglicht, mit einer geringeren Zahl von Ein-/Ausgangsanschlüssen des Steuergeräts 11 auszukommen. Dieser Gesichtspunkt ist insbesondere dann wichtig, wenn das komplette Steuergerät 11 auf einem Halbleitersubstrat integriert sein soll.

[0025] Als externer Bus kann im Prinzip ein beliebiger serieller Bus eingesetzt werden, der die im Einzelfall an die Übertragungskapazität gestellten Anforderungen erfüllt, insbesondere ein Internet-, Fire-Wire- oder USB-Bus. Besonders bevorzugt ist ein USB-Bus, u.a. wegen der unterschiedlichen Typen von Datentransfer, die ein solcher Bus standardmäßig unterstützt, und die sich jeweils für die verschiedenen, im Laufe des Entwicklungs- und Optimierungsprozesses auftretenden Übertragungsvorgänge eignen.

[0026] Das auf einem externen Computer implementierte Entwicklungstool 18 wird während der Entwicklungsphase des Steuergeräts 11 für diverse Anwendungen genutzt, z.B.:

- Debugging: Überwachen und Beeinflussen des

im Steuergerät 1 befindlichen Programms und gegebenenfalls Ändern von Befehlen des Programms;

- Messen: Übertragen der Inhalte von einzelnen Zellen des Speicherbausteins 3 mit einer der Häufigkeit ihrer Aktualisierung durch den Mikroprozessor 2 entsprechenden Häufigkeit an das Entwicklungstool 18 und Anzeigen des dem Inhalt der Zelle entsprechenden physikalischen Wert durch das Entwicklungstool;

- Verstellen: Verändern von von dem Steuergerät verwendeten Parametern, um die Auswirkungen der Veränderungen auf das Verhalten des von dem Steuergerät gesteuerten Motors zu untersuchen;

- Bypassen von Funktionen: Da die Rechenzeit und Ressourcen im Steuergerät 1 begrenzt sind, können Änderungen in der Programmierung des Steuergeräts dessen Zeitverhalten verändern. Um Funktionen zunächst ohne Rücksichtnahme auf die begrenzte Leistungsfähigkeit des Steuergeräts entwickeln zu können, kann die Berechnung von Werten einer Funktion im Steuergerät abgeschaltet werden, und die Eingangswerte der Funktion werden zum Entwicklungstool übertragen. Dieses führt die Berechnung der Funktion aus und überträgt anschließend die Funktionsergebnisse zurück ans Steuergerät.

[0027] Diesen diversen Anwendungen entsprechen unterschiedliche Anforderungen an das Zeitverhalten der Datenübertragung zwischen Entwicklungstool 18 und Steuergerät 11.

[0028] Eine erste Gruppe von Übertragungsvorgängen zwischen Entwicklungstool 18 und Steuergerät 11 sind die nicht zeitkritischen Übertragungen. Sie kommen hauptsächlich beim Debugging vor. Dazu gehören insbesondere die Übertragung des Steuerprogramms und der Ausgangsdaten, mit denen dieses arbeiten soll, in den Speicherbaustein 13 zu Beginn des Betriebs des Steuergeräts oder Änderungen dieser Daten während einer Unterbrechung des Betriebs des Steuergeräts 11. Für diese Art der Übertragung eignet sich insbesondere der sog. Bulk-Modus des USB-Busses. Im Bulk-Modus werden Datenpakete von maximal 64 Byte übertragen, und Fehler bei der Übertragung werden erkannt und behoben. Dieser Übertragungsmodus hat eine niedrigere Priorität als die anderen vom USB-Bus unterstützten Modi und kann daher unterbrochen werden, wenn in einem der anderen Modi Daten zu übertragen sind. Die geringe Länge der Pakete des Bulk-Modus erleichtert die Nutzung auch kleiner Lücken zwischen Übertragungsvorgängen mit höherer Priorität.

[0029] Um einen Fehler in einem Steuerprogramm zu finden, ist es häufig nötig, das Programm etappenweise ausführen zu lassen, in dem im Programmcode sog. Breakpoints festgelegt werden, an welchen das Steuergerät die Ausführung des Programms unterbrechen soll. Für die Übertragung der

Breakpoints an das Steuergerät und für die Rückmeldung von im Laufe einer solchen Etappe auftretenden Werten von Registerinhalten des Prozessors, Parametern etc. an das Entwicklungstool 18 kann ebenfalls der Bulk-Modus eingesetzt werden.

[0030] Bei Messanwendungen geht es darum, die zeitliche Entwicklung von Parametern des Motors zu verfolgen, die von den Funktionseinheiten 16 überwacht und an das Steuergerät 11 gemeldet werden. Diese Parameterwerte werden vom Steuergerät 11 jeweils an festgelegten Speicherplätzen des Speicherbausteins 13 gespeichert und können von dort aus gelesen und über den USB-Bus 19 an das Entwicklungstool 18 weitergeleitet werden. Um die dabei mit hoher Rate anfallenden Datenmengen an das Entwicklungstool 18 zu übertragen, wird vorausgesetzt der sog. Isochrone-Modus des USB-Busses eingesetzt. Dieser Modus verwendet Datenpaketete von bis zu 1.023 Bytes mit einer Frequenz von einem Paket pro Millisekunde. Da dieser Modus einen sehr geringen Overhead aufweist, sind große Datenmengen effizient übertragbar. Der für die Übertragung im Isochrone-Modus verwendete Code ermöglicht eine Erkennung von Übertragungsfehlern, aber nicht ihre Behebung. Wenn das Entwicklungstool 18 einen Fehler in einem im Isochrone-Modus übertragenen Paket feststellt, kann er, sofern genügend Übertragungskapazität zur Verfügung steht, eine erneute Übertragung anfordern; andernfalls muss das betreffende Paket verworfen werden.

[0031] Das Bypassing beinhaltet das Lesen des Inhalts einzelner Speicherzellen im Steuergerät 11 durch das Entwicklungstool 18 oder einen anderen an den USB-Bus angeschlossenen externen Rechner, hier als Bypassrechner bezeichnet, und das anschließende Ersetzen dieses Inhalts durch Beschreiben der Speicherzelle mit einem von dem Entwicklungstool oder dem Bypassrechner neu berechneten Wert. Dieses Bypassing kann z.B. eingesetzt werden, um in der Phase der Entwicklung und Optimierung des Steuergerätes 11 diesem gegenüber das Vorhandensein einer Funktionseinheit 16 zu simulieren, die in der tatsächlichen Anwendungsumgebung, für die das Steuergerät 11 vorgesehen ist, die an dem betreffenden Speicherplatz eingetragenen Werte liefern würde. Es kann auch dazu dienen, Rechenprozesse zu ersetzen, die nach Abschluss der Entwicklung des Steuergeräts 11 von diesem durchgeführt werden sollen, deren Programmierung in der Phase der Entwicklung des Steuergerätes 11, in der das Bypassing eingesetzt wird, noch nicht realisiert ist.

[0032] Ein aus dem betreffenden Speicherplatz gelesener Wert kann zusammen mit aus anderen Speicherplätzen gelesenen Werten im Isochrone-Modus an das Entwicklungstool 18 oder dem Bypassrechner übertragen werden; falls lediglich der Wert des betreffenden Speicherplatzes oder einer kleinen Zahl von Speicherplätzen an das Entwicklungstool 18 oder dem Bypassrechner zu übertragen ist, kann hierfür der Interrupt-Modus des USB-Busses einge-

setzt werden. Der gleiche Interrupt-Modus dient auch zum Zurückschreiben des vom Entwicklungstool 18 oder dem Bypassrechner berechneten neuen Wertes in dem betreffenden Speicherplatz. Im Interrupt-Modus werden Datenpakete von maximal 64 Byte bei 13 Byte Overhead übertragen. Fehler bei der Übertragung werden erkannt und behoben.

Patentansprüche

1. Kraftfahrzeugsteuergerät, insbesondere Motorsteuergerät (11), mit einem Prozessor (12) und einer ersten Schnittstelle (14) für die Kommunikation mit Funktionseinheiten (16) des Kraftfahrzeugs, dadurch gekennzeichnet, dass wenigstens eine zweite Schnittstelle (17) mit dem Prozessor (12) in einer Baueinheit vereinigt ist.

2. Kraftfahrzeugsteuergerät nach Anspruch 1, dadurch gekennzeichnet, dass die Schnittstelle (17) in der Lage ist, ohne Beteiligung des Prozessors (12) auf einen Speicherbaustein (13) des Kraftfahrzeugsteuergeräts zuzugreifen.

3. Kraftfahrzeugsteuergerät nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass die Schnittstelle (17) in der Lage ist, schreibend auf Code des Prozessors (12) in einem Speicherbaustein (13) zuzugreifen.

4. Kraftfahrzeugsteuergerät nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die zweite Schnittstelle (17) für den Blocktransfer von Daten ausgelegt ist.

5. Kraftfahrzeugsteuergerät nach einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, dass die erste Schnittstelle (17) mit dem Prozessor (12) und der zweiten Schnittstelle (14) in der gleichen Baueinheit vereinigt ist.

6. Kraftfahrzeugsteuergerät nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die Baueinheit eine Leiterplatte ist.

7. Kraftfahrzeugsteuergerät nach einem der Ansprüche 1 bis 5, dadurch gekennzeichnet, dass die Baueinheit ein Halbleiterchip ist.

8. Kraftfahrzeugsteuergerät nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass sie wenigstens einen Speicher (13) für Betriebsparameter des Prozessors aufweist, und dass der Speicher (13) über die zweite Schnittstelle (17) beschreib- und/oder lesbar ist.

9. Kraftfahrzeugsteuergerät nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die zweite Schnittstelle (17) eine serielle Schnittstelle ist.

10. Kraftfahrzeugsteuergerät nach Anspruch 8, dadurch gekennzeichnet, dass die zweite Schnittstelle eine Ethernet- oder FireWire-Schnittstelle ist.

11. Kraftfahrzeugsteuergerät nach Anspruch 9, dadurch gekennzeichnet, dass die zweite Schnittstelle (17) eine USB-Schnittstelle ist.

12. Kraftfahrzeugsteuergerät nach Anspruch 11, dadurch gekennzeichnet, dass die zweite Schnittstelle (17) in der Lage ist, Daten, die vom Prozessor (12) über die erste Schnittstelle (14) empfangen worden sind, im Isochrone-Modus zu übertragen.

13. Kraftfahrzeugsteuergerät nach Anspruch 11, dadurch gekennzeichnet, dass die zweite Schnittstelle Schnittstelle (17) in der Lage ist, Steuerparameter des Prozessors (12) im Bulk-Modus zu übertragen.

14. Kraftfahrzeugsteuergerät nach Anspruch 8 und Anspruch 11, dadurch gekennzeichnet, dass die Schnittstelle (17) in der Lage ist, einzelne Speicherstellen des Speichers (13) im Interrupt-Modus zu lesen und/oder zu beschreiben.

15. Kraftfahrzeug mit einem Kraftfahrzeugsteuergerät (11) nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die zweite Schnittstelle (17) mit keiner zu steuernden Funktioneinheit des Kraftfahrzeugs verbunden ist.

16. Verfahren zum Kommunizieren zwischen einem Kraftfahrzeugsteuergerät nach einem der Ansprüche 11 bis 14 und einem externen Host (18), bei dem der Host (18) für unterschiedliche Typen von zwischen ihm und dem Kraftfahrzeugsteuergerät auszutauschenden Daten unterschiedliche USB-Endpoints und Übertragungsmodi festlegt.

17. Verfahren nach Anspruch 16, bei dem der Host die Endpoints gemäß einer Prioritätsreihenfolge abfragt.

Es folgt ein Blatt Zeichnungen

Fig. 1

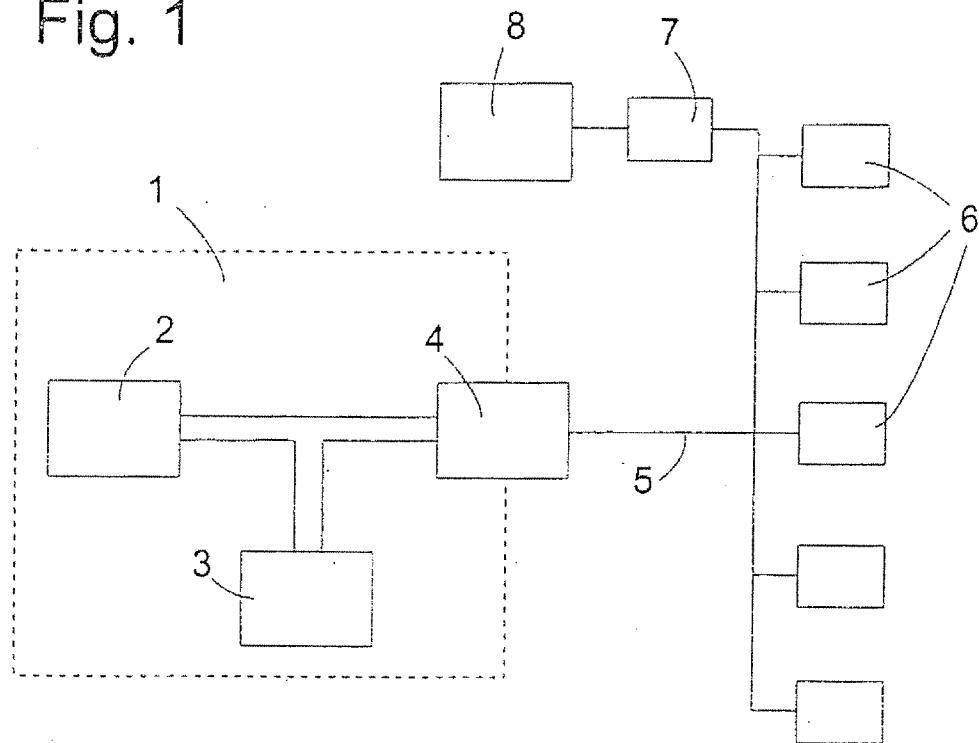


Fig. 2

